Mathematical Modelling for Software Engineering Students

Dag Wedelin

Abstract This paper describes the development of the course Mathematical Modelling for second-year IT/Software Engineering students at Chalmers. The course combines mathematical modelling and problem solving, exploring ways to teach these in the most efficient manner. Key elements are a large number of reality-oriented exercises, together with appropriate supervision sessions and follow-up lectures. Students rank the course among the best and most relevant courses in the curriculum.

1. Introduction

When I participated in the planning of this new program a few years ago, I came to think a lot about the fact that both in the existing D (computer scíence and engineering) and in the new IT (software engineering) programs at Chalmers, there was an unusually large distance between the mathematics courses and the other courses in the curriculum. In many engineering programs mathematics is studied in parallel with engineering courses where similar mathematics is immediately applied. At D and IT instead programming and other subjects are given side by side with the mathematics courses, and the link to mathematics is not so clear. Additionally, it is not uncommon that students have chosen in particular the IT program partly because they see mathematics as a topic on the side in their education. The result is that many D and IT students never understand the many links that actually exist, and also never learn how they can actively use mathematics in their profession.

Since it is difficult to predict in which areas different D and IT students will come to work, my solution was to suggest and create a broad course in mathematical modelling, rather than in some particular engineering subject. The course, named Mathematical Modelling IT, is at present compulsory in the second year of the IT-program. D students often select the course in their third year.

The course is given after the students have taken all the basic courses in mathematics (analysis, linear algebra, mathematical statistics and discrete mathematics), and in most cases also a first course in data structures and algorithms. The course does not focus on any particular kind of model or models, and does not aim to introduce more new mathematics. Instead, the connecting theme is mathematical modelling as a problem solving tool. The link to other areas within the IT program is then also easier to make.

Developing the students' ability in using mathematics has another important side effect in strengthening their identity as engineers. The reason is that many IT students are confused about what distinguishes their education from other apparently similar or related IT programs. One answer is that it has a mathematical/scientific foundation. But unfortunately this makes little sense, if the students do not themselves see the relevance of their mathematical studies, or feel confident in how to use this knowledge.

2. Mathematical modelling and problem solving

Given the general idea of a broad mathematical modelling course, the specific objectives for the course were defined from the perspective of long term usefulness, independently of where the student will actually work in future. The course therefore intends to give the student:

- Knowledge about different kinds of mathematical models, and how they can be used. Attention not only to classical mathematical modelling, but also to models common in computer science.
- Skill to use, create and evaluate mathematical models in different and possibly new areas of application.
- Significantly improved skills in general mathematical problem solving.
- Perspective on the role of mathematical modelling and mathematics in general, and for the professional engineer.

To understand how this relates to the general aim, the way I see mathematical modelling is that it is the first step of applied mathematical problem solving, i.e. the actual link between reality and theory, see figure 1. Knowledge of mathematical modelling is therefore enabling knowledge to apply mathematical methods in any given area. The skill of mathemtical modelling can be seen as a specialized form of mathematical problem solving.

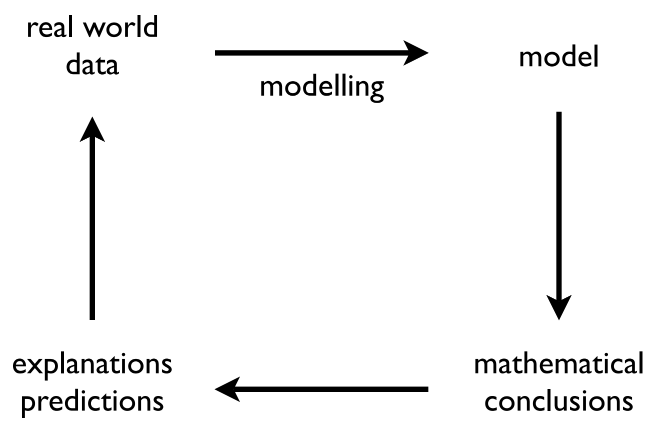

Figure 1. The modelling cycle.

Turning to general problem solving, my primary concern is not the straightforward application of given methods, such as solving a quadratic equation. Rather, I refer to the ability of more complex problem solving such as making suitable assumptions and creating a model, and tasks that include planning to reach a non-trivial goal in several steps, setting subgoals, and the use of some systematic strategy and search to accomplish this. Pòlya [4] refers to this as structured problem solving. In terms of Blooms taxonomy one can roughly think of this as the three highest levels. It can be added that this is an area where many students have very limited experience, which is another reason for their difficulty to see how mathematics can be used in practice.

So in simple terms, the course can be described as 1) adding a critical missing piece of knowledge (mathematical models), and 2) help the students to develop the skills to fully use their mathematical knowledge (problem solving). See figure 2 for a comparison with Blooms taxonomy (revised version by Anderson). An alternative name for the course could therefore be “Mathematical modelling and problem solving”. As a result, this course together with the mathematics the student already knows, gives the student a considerably more complete set of tools to tackle unknown problems in new situations.

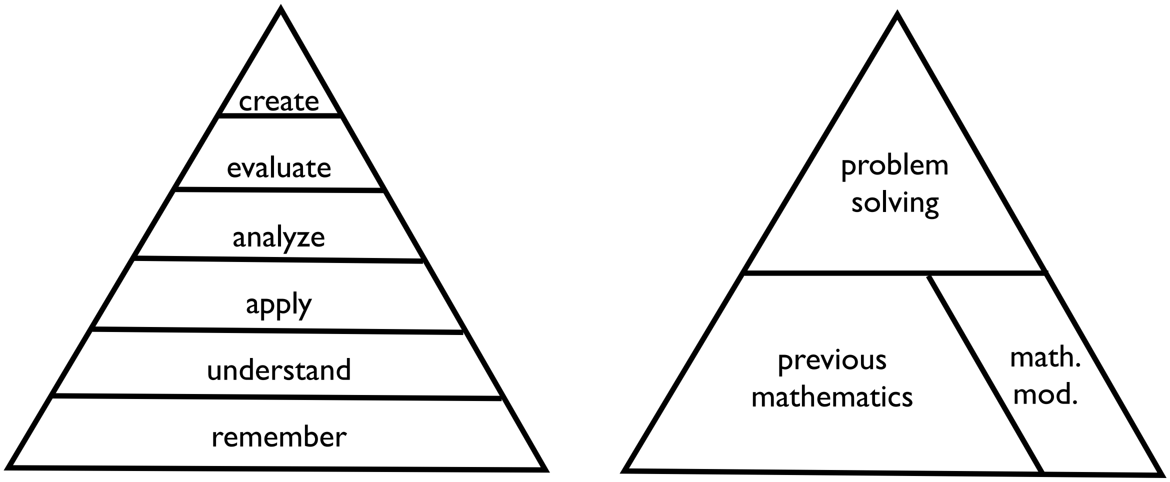

Figure 2. Previous mathematical knowledge, mathematical modelling and problem solving.

3. Course overview

We will now consider how the course was designed to meet the course objectives. The most important decision was to make a large number of exercises the core of the course. This works well to show different situations where mathematics can be used, and is also natural given the strong emphasis on improving non-trivial skills. To emphasize this, and to make the task feasible within the scope of the course, there is no course book. Also, the amount of new mathematics in the course is kept to a minimum.

The main structure of the course is shown in figure 3. It is organized in weekly modules, each considering a main class of models. A module begins with an introductory lecture where the module is presented. The students are then to work with the exercises during the week, in groups of two. During the week supervision sessions allow students to work in rooms where the can get get interactive help along the way. Solutions are handed in on Sunday at the latest. The total time to spend on the exercises is about 20 hours, and the number of consulting hours is about 10, during which all teachers work in parallel. Later in the next week, there is a follow-up lecture giving feedback on the exercises and other perspectives on last weeks module. In the next module students are also asked to make a short reflection on what they learned in the previous module. There are also a few introductory and final lectures. At the end of the course, all students are requested to write an individual final report about the course. (For more information

This structure has some general advantages:

- The main types of mathematical models are clearly reflected in the stucture of the course.
- Students are motivated to attend the first lecture since it is immediately related to something that they have to work with and hand in by the end of the week.

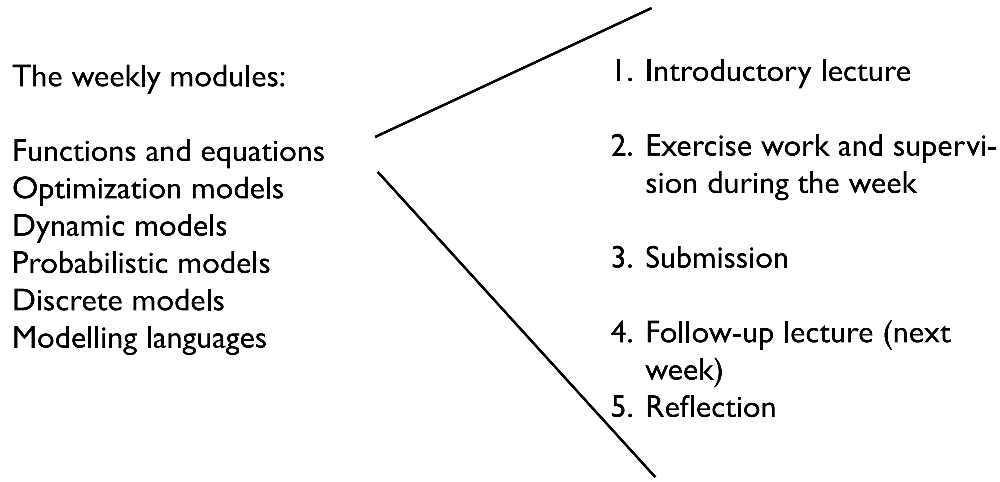

Figure 3. The weekly modules and their structure.

- Students start working immediately, and continue throughout the course. Since they get a non-trivial task to work with right away, we can also quickly identify groups that require additional support. A consequence is that at the end of the course we have an unusually high proportion of passed students.

- The high degree of interaction with the students makes it possible to get to know many students, to continuously give individual advice and feedback, and contributes to a good working atmosphere.

- Students will practice to write and explain.

- The follow-up lecture can start on a higher level, since the students already spent a lot of time with the exercises, and efficiently gives feedback on this work before the answers for the next module are handed in.

- The final report gives students an opportunity to structure and reflect on the course and their own work, and gives a lasting and concrete result that they can keep. It also provides a rich source of information for assessing the individual student.

First and foremost however, the structure has been set up to support and accelerate the learning of the skills of modelling and problem solving, and these different teaching elements all contribute in their own way to reaching this goal.

As an example illustrating the main teaching philosophy of the course, we can consider one of the first exercises, where the students are asked to find a curve that fits a number of points given in a table, see Figure 4 for a plot. No systematic method for doing this has been presented, and many students are confused about the exercise. However, by asking the students who are stuck if they have made a plot of the points, most begin to see how they might work, although it happens that some draw so carelessly that they start making false conclusions and are lost. After the plot those who so need are asked if they can think of examples of different functions. A common student question is "is it allowed just to do that?". By observing the curvature of the points, many students hypothesize the logarithmic function. When they ask for confirmation, we ask back what they think the value of the function should be in 0 and infinity. They then usually discard their hypothesis by themselves. Another common student question is "is this good enough?", to which we ask if they are satisfied. Some students do some polynomial approximation that they get out of Excel, giving a very good fit. In the follow up lecture, a simple formula with a excellent fit is presented and different ways to find it are discussed. We also consider how other alternatives can be discarded, or why they are not preferable. They also learn that what they have - or could have - discovered, is in fact one of Keplers laws. This gives an opportunity to give a short historical background and reflect on the nature of scientific discovery. The exercise also paves the way for a later exercise where the least squares method is introduced.

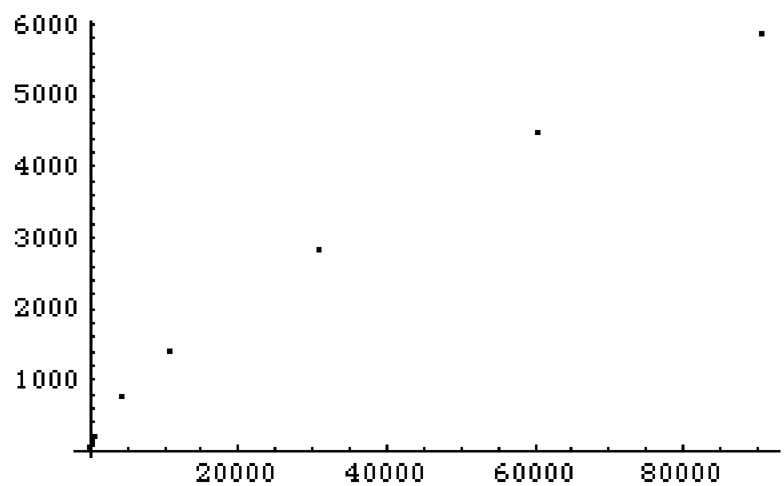

Figure 4. Find a function that fits the points.

It is also important to consider how the learning of problem solving can be supported in the assessment. When we assess the answers of the exercises we do not require that the answers are correct, and it is sufficient with sincere problem solving attempts. This helps the students to focus on how to solve problems, rather than on the solutions, and also to dare to be creative, especially since many students seem to percieve mathematics as something as something very formal. In also makes it easier to accommodate the large differences in the abilities of different students.

One common element of teaching that has not been used (except for the fact that students work in two person groups), is to encourage or even allow group work. If the purpose is to solve a given task quickly, this is excellent, but the objective of this particular course is more like learning to play a musical instrument. Then everyone must first practice on their own, to improve his/her own skills, and eliminate individual weaknesses.

It would be possible to discuss in groups after solutions have been handed in, and some form of organized peer feedback could be considered, although this has so far not been implemented. We do however strongly encourage students to talk to each other and discuss after they have handed in their answers. Group work generally is not unusual in other courses on the IT-program, so they still have the opportunity to practice this.

For more detailed and up to date information about the course, please also see the course homepage [1].

4. Defining the course with exercises

When designing the exercises, I had a general aim to consider situations that the student may encounter in the future. I have tried to make this explicit by listing some *general pedagogical considerations* that strongly influenced the design and style of individual exercises:

- **Realistic**. The exercises should connect to reality, to create starting points that students can understand and relate to, and to demonstrate how different methods can be used in practice. This does not mean that they need to be fully realistic, but rather that they have a sense of realism.

- **Exploratory**. The exercises are formulated so that solving the exercise becomes a form of exploration. There should be something sensible or interesting to discover (beyond say a numerical value without any real significance).

- **Challenging**. The exercises should be sufficiently difficult to force the student into real problem solving, and formulations may be incomplete and require additional interpretation and assumptions. They are intentionally designed to be too difficult to be solved by the students themselves, and are intended to be solved in interaction with the teachers. In this way, abilities are tested and individual weaknesses are revealed and discussed. The number of exercises, their difficulty, and the level of detail given in the exercises, have been tuned over time to correspond to what most students are able to handle.

- **Motivating**. This is already satisfied on a basic level by making exercises realistic to the extent that the student can appreciate or at least accept the exercises as sensible. Then they learn because they have a problem to solve, and directly see why certain knowledge is useful. The exploratory style of the exercises also make students curious about the answer. Other ways are to have problems with surprising answers, or which lead further than expected.

Then, a broad range of exercises is needed to cover different problems that require different approaches and perspectives. Note that this does not imply surface learning since we use the exercises to learn skills on a deeper level, and the differences between exercises enable important distinctions and comparisons. One way to define the course could be simply to present a number of randomly selected problems, but in this course the selection and combination of exercises was done so that they together also ensure a given degree of completeness. My approach to this was to explicitly consider the following *content-related components*:

- **Knowledge**. To meet the knowledge requirements I started out with a something that looks like a table of contents for a course in mathematical modelling. Different kinds of models such as functions and equations, optimization, dynamic models etc., their possibilites and limitations, and how they are applicable in various applications. Specific methods like analysis of scale, least squares, simulation etc. But also varying areas of application. Some exercises are IT specific, but not more than to keep the course relevant for other programs where the relative weight of mathematics is similar.

- **Skills**. Similarly, different skills of modelling and problem solving need to be present in the exercises. We can at least attempt to list these in another table, although they are partly tacit and thus difficult to make explicit. The exercises need to be sufficiently open to mimic real world situations where the first step is to understand the problem and make suitable assumptions. To make this possible, we naturally must not in advance give specific procedures for how to solve problems of different kinds. If we want the students to learn problem solving, many of the exercises must be of that character that they require several steps to be solved.

- **Perspective**. For example, a perspective on modelling includes some understanding of where and how exising models are commonly used, where they come from, to illustrate unexpected possibilities and limitations of some modelling approaches, to constrast simple inexact models against complex and more precise models, how different models can be used to solve the same problem. A perspective on problem solving includes not to expect that a solution can be found directly, appreciating the importance of really understanding a problem, that a solution is often presented in a way that is quite different from how it was concieved, the nature of scientific discovery, and so on. Within perspective I also include different kinds of reflection, for example on your own problem solving.

One can then see the design and selection of exercises as an optimization problem, so that the entire set of exercises should to the greatest degree possible satisfy several objectives. If single exercises serve multiple purposes, it may also possible to learn more in less time. The knowledge component is probably most apparent if you simply read the exercise, what skills and perspectives that can come out of them is usually more hidden. Students are therefore often only partly aware of what they are expected to learn in each exercise, which I do not consider as a problem. An observation is that it seems to be easier to design a single exercise that combines these different components, than designing an exercise that combines several aspects of the same component.

As mentioned above, it is difficult to meet all objectives with only a few large exercises. Therefore each weekly module consists of 4-8 exercises. It then becomes possible to match the knowledge requirements so that every listed piece of knowledge is present in at least one exercise, and similarly with the skills. A consequence of this way of selecting the exercises is that they will exhibit a considerable degree of variation. Another consequence is that you must be careful not to replace an exercise with another without considering the ensemble of exercises, which would be possible if exercises were only support for understanding the knowledge provided in the lectures. We can note that this is not a course with any free choice in what you will do, but rather the structure ensures that the intended completeness is met.

Although not every exercise can be designed to meet all criteria, it clearly makes a difference to have that ambition. It also makes a difference how the theme of an exercise is used, even if it is quite unremarkable. For example, we use the curve fitting problem to place the student in a problem solving situation where something is to be discovered by exploration, rather than using it as confirmation of a method that was already presented in a lecture or in a book. Therefore the student's perception of the exercise is different, and other skills will be learned. Clearly however, this perception depends on the prior knowledge of the student, which is different beween individuals. A few exercises may therefore be on the simple side for some, but many exercises are more open ended, and I also provide some optional more difficult exercises.

As an example with an exercise with a surprising answer, the students are asked to calculate the expected time of travel between two cities in a given road network. The question is then to see how the expected travel time between two cities improves if an bridge is built, in addition to the already existing roads. However, it turns out that the travel time increases. This, in addition to the problem solving required to find the solution, gives perspective on the relation between mathematics and common sense, as well as of the qualitative value of simple mathematical models.

One programming exercise is included in the course. The task is to create a graphical two dimensional simulation of two balls of different size and weight, that move under the influence of gravity, and which also may colllide with each other. The task is relatively straightforward, but requires some recollection of basic mathematics, a systematic approach involving defining several subgoals, and attention to details such as the definition of angles and reference directions. Some students have trouble because they are not careful with details, and learn a lot from that experience. Even without such problems, the exercise gives some perspective on the relation between pure mathematics, and the additional considerations required when you actually want something to work in a computer program such as a simulator. Many students patch mathematical errors with additional code, and are amazed when they realize that with the correct math all problems magically disappear.

For many exercises Mathematica is useful, and it becomes a part of the course to become acquainted with this system on an basic level.

In the literature on mathematical modelling, many books consider classical physics, mathematical characterizations of different solutions to differential equations, or various special areas of application. I found two books that inspired me, one by Giordano [2] that had the kind of completeness that I was looking for, and Meerschaert [3], that had organized the parts of the book in a way that influenced how I chose to organize the exercise modules.

5. Teaching modelling and problem solving

For some students, it might be sufficient simply to give them the exercises and then let them work on their own. However, in the course we teach and actively support the learning of modelling and problem solving is several different ways.

Modelling and problem solving strategies

At the beginning of the course we present some general problem solving strategies and hints both for modelling and generally. One such template for problem solving is the following (freely adapted from Pòlya [4]):

1. Understand the problem.

2. Plan the steps of your problem solving.

3. Be careful when you carry out your plan.

4. Check your results and conclusions.
   Reflect over the solution process.

We use other templates for different kinds of modelling, and also give other general hints that can be useful (Always do something, never stop. You can always try to understand the problem better. Draw a figure. Solve a simpler problem…). While all these somewhat abstract instructions may have little meaning to the students in the beginning, this changes as they work with the exercises. In fact, when you consider what you need to solve different problems and see how many students work, it is striking how relevant the general strategies are.

Supervision

We do not expect that most students will be able to solve the problems by themselves. The supervision is a very important part of the course, in which all teachers are engaged, and where we seek a continuous dialogue with the students.

During the supervision sessions the students sit in classrooms or computer rooms and the groups (two persons) can obtain help from the teachers while they are trying to solve the problems themselves. Students receive attention mainly if they ask, but we also try to walk around and follow what each group is doing. However, we avoid giving direct answers. Instead, the idea is to supervise mainly by asking questions. This is done in an interactive style, where we in every situation try not to give more help than needed, and if this help is not sufficient, they can get more help if they ask again. In this way we avoid that the students are totally stuck, and we can hopefully accelerate learning and provide specific help to each group.

While some students seem to get along very well from the beginning, many have very little experience with structured problem solving. Instead, they have a habit of one step problem solving, where the assumed task is to match the current problem with the closest available formula or method. An important part of the supervision therefore becomes to convince the students not to rush to the solution, that the problem cannot be solved in a single step, and remind of general problem solving strategies.

During supervision, it is important to recognize and respect that it is very frustrating for the students to see that approaches that they have previously applied are inadequate for the problems that they are now facing. So we also need to encourage the students and convince them that they are capable of solving the problems, if they just learn to work in a different way.

A comment that I occasionally give during supervision is that "I asked some questions, and you gave me the answers. So, you could have done it yourself!". This usually leaves the students a bit confounded. At the end of the course this is taken a step further. This is because also the style of supervision has more than one purpose. The first and obvious purpose of the supervision is to assist the students in their problem solving. But another purpose is to illustrate a method of internal dialogue so that they in the future can supervise themselves. In the last week we have a short exercise about this, and it is also discussed in the final lecture.

Finally, the many hours spent in supervision gives rich opportunities for the teachers to see how the students work, and to give feedback and informally assess their work. It turns out that relatively open and challenging exercises that the teachers know well, and a lot of interaction, is also very effective for discovering misconceptions and substandard working practices. Many students see formulas as given once and for all or think of mathematics as something where form is more important than content. With respect to working practices, many students have never learned the importance of being careful, or do not see the importance of really understanding what they are doing. Some also program in a rather stochastic way, wildly guessing and testing in the hope of getting things right. A less serious issue is that some students must learn to better plan their time, and set their own limits and standards when encountering open assignments. It is notable that in many cases students are themselves not aware of their problems, because they have previously not encountered situations that reveal them, or because they have not themselves understood the nature of their own current limitations.

Follow-up lectures

In the follow-up lecture for each module we give feedback by going through the exercises and propose good solutions, different apporaches to solving them, as well as common errors or problems that the students had. Thanks to this lecture, it does not matter so much if the students do not reach all the way to a solution of the problem. When they have spent time struggling with the exercises, also the weaker students are in a good position to appreciate the solutions, and maybe even learn more than those who were more successful. It also enables every student to reflect over the difference between their own attempts, and the solutions I present. I encourage this by requesting everyone to ask themeslves the question: "What was it that prevented you to find this solution by yourself? If you find the answer, you will become a better problem solver".

The second purpose of the follow-up lecture is to continue beyond the exercises, to give perspective and draw general conclusions about mathematics, modelling and problem solving.

From an organizational perspective the follow-up lecture is an efficient way to give feedback to a large group of students. After the follow-up lecture, students are encouraged to talk to us individually about any still outstanding issues. They are also asked as a part of next week’s module to briefly write down their reflections on what they learned from the module.

6. Other considerations

Communicating the objectives of the course

When the course begins, there are several challenges in the communication of the objectives and practices of the course, since the way this course works is different from what the students normally expect:

- **The purpose of an exercsise**. Many students expect mathematical exercises to have some easily understood single purpose, such as practicing a given method, and with a single right answer as a confirmation of success. In this course however, the purpose of the exercises is only to a limited degree to find the best solution, if it even exists. More important purposes are to illustrate some area of application, and to learn problem solving by working with the problem. For example, an intended part of the exercise is to understand it, but students may initially feel that a question is fuzzy or ill-defined, and see this as a sign of misjudgment or lack of preparation on my part. It is also important to explain that we allow all students to make mistakes, in order to develop their creativity, and that we see this as perfectly normal.

- **Interaction**. Some students expect courses to be set up so that students and teachers should not need to talk to each other. With respect to supervision, it must be emphasized that students can get more help with an exercsise every time you ask. Otherwise, students who finally dare to ask about an exercise, just get a single question back and leave with the conclusion that the supervision was useless. So the interactive nature of the course must be very clearly communicated.

- **Discipline**. Students are clearly instructed to keep their problem solving within the groups. However, after a module has been handed in, they are encouraged to talk to each other. Note that this is not an issue mainly wih respect to examination, but a pedagogical necessity to give every student the opportunity to learn problem solving. Some exercises are especially vulnerable, in that a single word may reveal an hidden answer. However, it is also sometimes necessary to remind single students that all exercises are part of the examination, and are not just a voluntary preparation for an exam.

- **Grading**. Initially, there is also some concern about what to expect from the course and especially the supervision. Some students are worried about not being able to solve the exercises, or know exactly what is expected of them to pass or for a particular grade.

However, after having calibrated the message over several years, any initial confusion usually fades away quite quickly, and after a little more than a week almost all students have a good understanding of the intentions of the course. The main difficulty where I did not fully succeed during the first two years of the course, was to impress upon the students the importance of really attending the supervision. The result was that some students spent many hours by themselves, struggling with exercises that were never intended to be solved in solitude.

Grading and examination

To pass the course, all exercise modules and the final individual report must be accepted. To pass an exercise module, the most important requirement is that the group has seriously attempted to solve the exercises, but they are not required to find any final or correct answers. We also require basic facts to be correct, and an acceptable presentation. For passing the final report a basic understanding of the described model types, the modelling process and basic problem solving is required, as well as an acceptable presentation.

In practice these requirements mean that if a student agrees to participate in the course, and spends a reasonable number of hours with the exercises, he/she can feel confident to pass. This is important to make the students relax and take interest in the real challenges of the course. Otherwise, it would be very stressful to present open and difficult exercises, that may go beyond what a student is able to solve. It also reduces the motive for cheating.

For a higher grade we also require a certain absolute quality level for the answers to the exercises, and also for other criteria such as demonstrating a deeper understanding of methods, own reflections and a high quality presentation. Similar criteria are applied for the final individual report, which is a good source for assessing the depth of knowledge especially since we ask not just for facts but also for reflections. The final grade is weighted between the exercises and the final report, approximately by a ratio of 2:1. To counter the observed effect that some good students use this ratio as a tactic to minimize their effort in the final report, I have added that I normally do not give a higher course grade than that given to the final report.

Clearly there is an inevitable element of subjectivity in the grading. To handle this, I encourage students that are not happy with their grade to come back to me, and if a higher grade is at all in reach I give them hints on how they can improve or extend their report. This seems to work well.

A practical problem is that the reading of all the exercises is time consuming, and together with all the supervision it is easy to break the course budget. This year (2008), we experimented with a randomized scheme where only some exercsise modules were graded (takes a long time), and the others were only scanned to find those that clearly did not pass (much faster). In this way we received a more reasonable time distribution for the course assistants, who could spend about 50% of their time in supervision directly interacting with the students.

One issue that has been frequently commented by students is that of individual comments on the exercise modules. This is simply not possible given our budget. On the other hand, I personally think that the system with the feedback lecture is very efficient not only in terms of resources but also pedagogically. We also encourage anyone that wishes feedback beyond the feedback lecture come to us personally for a discussion.

Since a lot of effort has been put into creating the exercises, it is not possible to replace exercises from year to year, which unfortunately creates an opportunity for cheating, so special attention must be taken in distribution information about the solutions of the exercises.

Continuous improvement

During the years that the course has been given I have worked with a systematic method for continuous improvement. This involves immediately making notes of all own ideas or comments that occur while the course is given. I also continuusly collect comments from students and course assistants, and the course evaluation. In this way a large number of items are collected, and before the course begins the next year I systematize these and make as many changes as possible.

Other major changes of the course have included a revision of all the lectures, and a manual for course assistants, giving general instructions for supervision as well as specific comments and hints for most exercises. I have also continuously in a somewhat experimental way improved the way we communicate the objectives and philosophy of the course, and well as how we grade the exercises.

Another development is that I have over time come to increase the emphasis on general problem solving, rather than mathematical modelling as a particular subject.

7. What do the students think?

Through the combined sources of exercises, final reports and course evaluations, there is extensive documentation of how the students view the course. Some comments have been collected and are available from the course homepage. One student writes (2008):

> *For the first time during my studies I was able to connect the course both with previous courses as well as with the kind of problems I can expect in my profession as an engineer. The course has mainly given me the following:*
>
> 1) *Model thinking. Common models in mathematics and in practice. Prior to this course, I do not think I had a clear formal understanding of what a model is. Such an understanding is very important for working with models, and for seeing their limitations.*
>
> 2) *Problem solving. A structured view of problem solving. Problem solving is something that everyone can do to a certain degree, but how to approach a problem in a structured way is more difficult to define. I feel that this course has provided me with a standard path to follow in problem solving. In the reflections that were part of the course, I also could look back on the previous week, analyze my approaches to see what went wrong, and see how I can improve. I have realized that my main weaknesses are that I am not careful enough, I give up too easily, and that I must attack a problem from different angles to reach an answer. Problem solving is the most important skill of an engineer, and I have appreciated this opportunity to practise and develop this skill.*
>
> 3) *Practical application of mathematics learned in other courses. This is the last compuslory mathematicats course I will take here at Chalmers. I, and my fellow students, have acquired a large mathematical toolbox, and in this course you learn to make an inventory of this toolbox, and how to select the right tool for a given problem.*
>
> 4) *To see the whole picture, and not just focus on the mathematically correct solution. How will the solution change if some assumptions turn out not to hold?*

Many students express a fundamental change in their abilities, in their perspective on mathematics, and its role in their future profession. Comments like "*I have never before thought about were the formulas come from*", and "*the course has lifted me to an entirely new level*" are common. Students clearly convey that they see the course as an important part of their education and many express gratitude for having been given this opportunity. The course is viewed as the best mathematics course in the curriculum or even overall. Some students express that their experience with mathematical modelling will also help them to be better programmers. Another not uncommon comment is to ask about more courses like this, and they also wonder why this did not happen earlier in their education.

The course is considered to be demanding with all the exercise modules, but the percentage of students that pass the course more or less in time is also unusually high compared to other courses (90-95%). However, we monitor the number of hours that students work on each module, and it is unusual that the effort exceeds the hours that are officially assigned to the course.

The question about personal written comments to the exercises has almost disappeared by better explaining the importance of attending the follow-up lecture, and by encouraging students to talk to us and get additional comments after that.

In the most recent course evaluation (2008) the grade was 4.6 for relevance (average grade on a scale from 1-5), 4.6 for pedagogy, and the overall course grade was 4.5. In a survey by the IT program adminstration including all courses in the first three years, this is the highest overall grade.

8. Conclusions

The course has now been given for five years (2008) with about 100 students per year. In the last years, not only IT students, but also many D students have selected the course.

As I see it, the single most important contribution of the course is that it exists. Giving a course like this in a D or IT-program is unusual and is not part of established recommendations for the software engineering curriculum. Instead a more computer-centric view is usually taken in deciding what kind mathematics should be taught, and the role of the software engineer is often defined in quite specific terms. In contrast to this, this course sees both the computer and mathematics as important general purpose tools for the IT-engineer, and provides the basic skills for this to become a reality.

Another characteristic of the course it its broad view on mathematical modelling, with a balance between continuous and discrete mathematics adapted for the intended audience. We note that while it is certainly possible to teach mathematical problem solving independently of mathematical modelling, the combination of mathematical modelling and problem solving is useful pedagogically, since the modelling part provides motivation with real-world oriented exercises, and the problem solving skills are then required to solve them.

Finally, the course has an ambition not just to practice but to actually teach non-trivial mathematical problem solving. I think it is fair to say this is generally considered as difficult, and it is often ignored when teaching mathematics generally. A considerable pedagogical effort has been required to meet this objective in a course with many students, and within the budgetary constraints of an ordinary course. Here, the factors that I consider as especially important have been to build the course around the exercises, the interactive style of supervision by asking questions, and and not to have any coursebook.

When comparing the pedagogical approach to other courses, one might say that the course represents an intermediate step between straightforward knowledge oriented courses, and project courses. Compared to traditional courses this course has a much higher emphasis on methods and skills for handling the unknown. On the other hand, it is a common expectation that students will automatically learn problem solving skills in project courses, which in my opinion is often inefficient. With a single large project (and often with tasks split between students), the student will not encounter the wide range of problems and approaches that can exist, and thus will not learn to make appropriate distinctions and choices, or even know what to expect. It may also be difficult to observe and efficiently improve the skills of individual students, especially if the task of the project is open and not well known even to the supervisor. Clearly project courses, essays and thesis projects are very important, and they are in a sense the final stage. But the controlled setting of this course, with a large number of varied problems that the teachers know very well, can be seen to fill a gap also in this pedagogical sense.

Personally, I consider the course to have met and in some aspects exceeded the objectives that were set up from the beginning. My basis for this conclusion is mainly the continuous dialogue with the students during the course, where improvements are clearly visible. The final reports, and the overall response and feedback from the students are also important sources.

It can be noted that the skills that the course intends to convey are open-ended. To develop your problem solving is something very personal, and with challenging and open-ended tasks, some students will go very far. Those who are on a more basic level, will still see how different mathematical methods can be used to solve real problems, and learn some important basic skills. So students begin at different levels, and end up at different levels, but all will have improved.

The experience from the course confirms that many of the students on the IT program have little experience even in very basic mathematical modelling and structured problem solving, and are therefore crippled in how they can exploit their mathematical knowledge. Many have an unclear view of what skills you really need, what you will be able to do in practice if you possess such skills, and generally what can be expected of them as engineers. The perceived powerful effect of the course is then achieved by the combination of supplying two critical missing pieces (mathematical modelling and problem solving) through reasonably realistic exercises linking to previous mathematics, and also helping students to correct misconceptions and working practices.

In retrospect, the decision to make the course compulsory was important since students would otherwise not have any idea what yet another mathematical course could do for them. It would also not have given the strong signal that mathematics is an important problem solving tool also for IT engineers. Some concerns that the IT students would not appreciate the course turned out to be unfounded. Another comment I received at an early stage was that a course like this was unnecessary because it could be integrated in other mathematics courses. This may be possible, but as long as these courses do not change radically, both the mathematical modelling and the problem solving perspective would easily be overshadowed by the struggle to learn the new mathematical theory.

A possible disadvantage of a course like this is that it is not so easy to reproduce or hand over the course, other than working side by side with people that can continue in the same spirit. This is a challenge since the organization may not allow or promote that. The course assistants must be well prepared, preferably by solving many of the exercsises themselves before they supervise, and they must understand the teaching philosophy of the course. The attitudes and communication skills of the course assistants also become very important. It may be difficult to write a coursebook corresponding to the course.

While this course was designed to meet certain objectives that I was able to choose freely, other courses have other objectives and may be constrained by their role in the curriculum. However, I will point out a few aspects that I have found to be generally useful, and that I would myself consider to use also in other courses:

- Design a course by considering what the student needs to know and be able to do in the long term, and then systematically develop the course from this without regard to tradition.
- Make sure not to forget to teach important things even if it is not in the title of the course.
- Use a collection of exercises as the starting point. A completeness in the knowledge requirements can be achieved by a sufficient number of smaller exercises. Then you can start from an ordinary, knowledge-based course description. By doing something first, students are activated, you get to know their ability, and it is also easier to motivate theory.
- Exercises are realistic, exploratory and have multiple purposes (knowledge, skill, perspective).
- The interactive and interrogative style of supervision is very effective to teach problem solving and to reveal problems in prior knowledge. Regular feedback and reflection is important.

As a final remark, I feel that my experiences from this course have made me reflect more over how we teach mathematics, as well as over the general priorities we usually give to knowledge, skills and perspective in the curriculum. Should it even be possible to give a single course that is considered as so important by the students?

Acknowledgements

I wish to thank my course assistants Birgit Grohe, Peter Gennemark, Carlos Gonzalia, Mikael Onsjö and Torbjörn Karfunkel who have all worked in the course and constructively contributed to what it is today. Special thanks to Birgit Grohe who was in the course from the beginning and contributed with a lot of improvements to the exercises and to the course as a whole. I also wish to thank all the students for the overwhelming response we have received for the course.

References

[1] Course home page at www.cs.chalmers.se/Cs/Grundutb/Kurser/modell/ (or via my personal page)
[2] Giordano et.al. A first course in mathematical modeling. Cengage Learning, 3rd edition, 2002.
[3] Meerschaert. Mathematical modeling. Academic Press, 1999.
[4] Pòlya. How to solve it. Princeton Universit Press, 2nd edition, 1957.